B2 98P1671

(12) UK Patent Application (19) GB (11) 2 315 964 (13) A

(43) Date of A Publication 11.02.1998

(21) Application No 9716206.9

(22) Date of Filing 31.07.1997

(30) Priority Data
(31) 08200874 (32) 31.07.1996 (33) JP

(71) Applicant(s)
NEC Corporation

(Incorporated in Japan)

7-1,Shiba 5-Chome, Minato-Ku, Tokyo, Japan

(72) Inventor(s)
Shinichiro Iwata

(74) Agent and/or Address for Service
Mathys & Squire
100 Grays Inn Road, LONDON, WC1X 8AL,
United Kingdom

(51) INT CL^{6}
H04L 29/06 , H03M 7/40 , H04L 1/00

(52) UK CL (Edition P)
H4P PENX

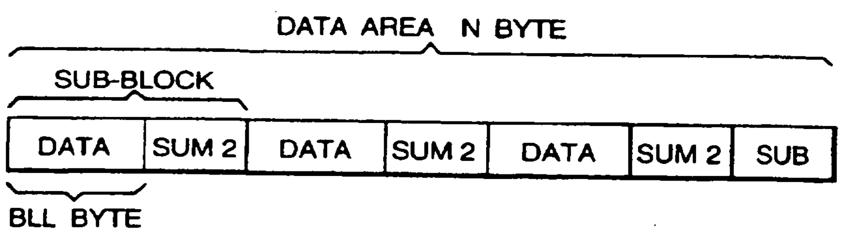
(56) Documents Cited
EP 0503768 A1 US 5410308 A US 5079548 A

(58) Field of Search
UK CL (Edition O) H4P PENL PENX
INT CL^{6} H03M 7/40 , H04L 1/00 1/08 1/18 29/06
ONLINE WPI & JAPIO

(54) **Transmitting fixed length data blocks containing variable length sub-blocks**

(57) A data communication system comprises a transmitter for transmitting a fixed-length block of information, and a receiver for receiving the block. The fixed length block includes a plurality of sub-blocks, each having a variable length depending upon the data. When an amount of transmission data has a size less than that of the fixed-length data block the transmitter repeatedly arranges the information within the same fixed-length block in order to form the block. Thus the unused portion of the fixed-length block is effectively utilized to increase the reliability of the data communication. Additionally, error check codes may be stored within the data area to further increase reliability.

Fig.5.

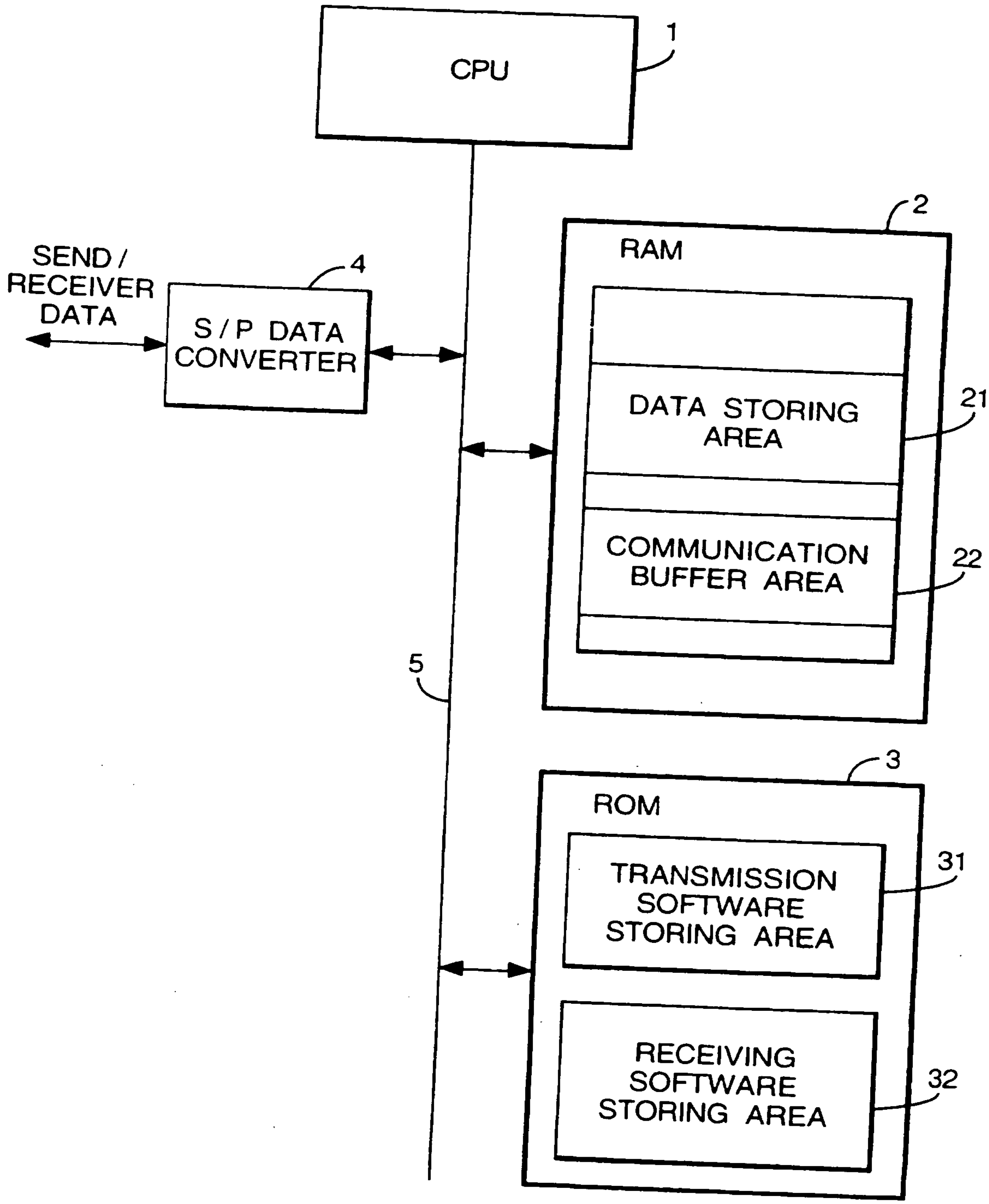


GB 2 315 964 A

At least one drawing originally filed was informal and the print reproduced here is taken from a later filed formal copy.

BNSDOCID: <GB___2315964A__I_>

Fig.1.



OCID: <GB___2316964A__I_>

Fig.2.

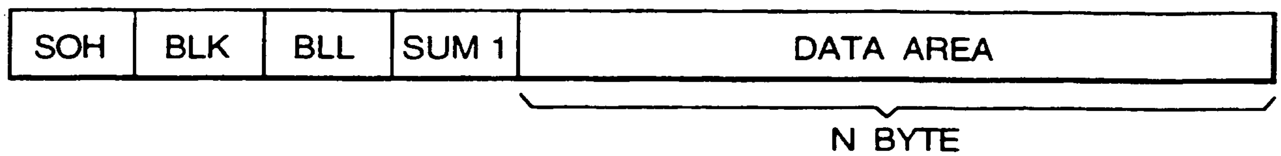


Fig.3.

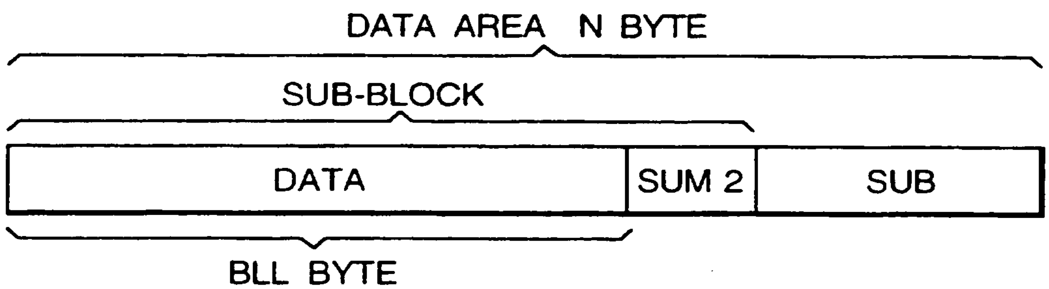


Fig.4.

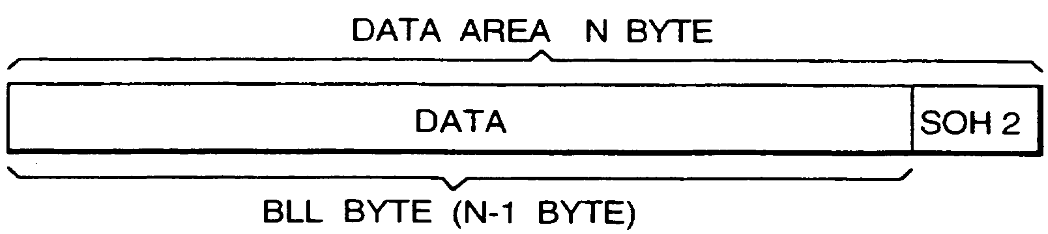


Fig.5.

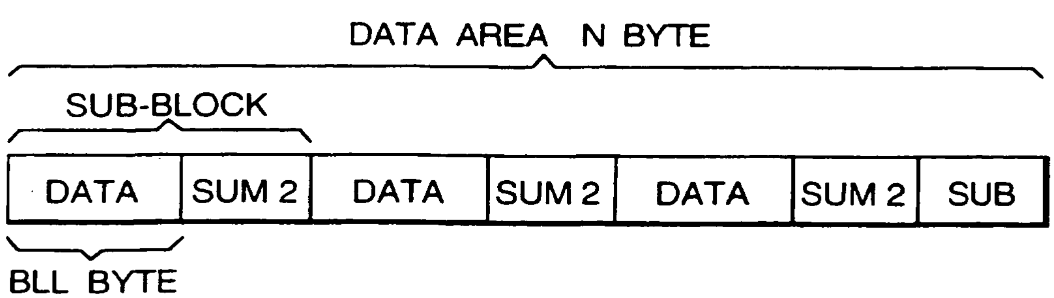
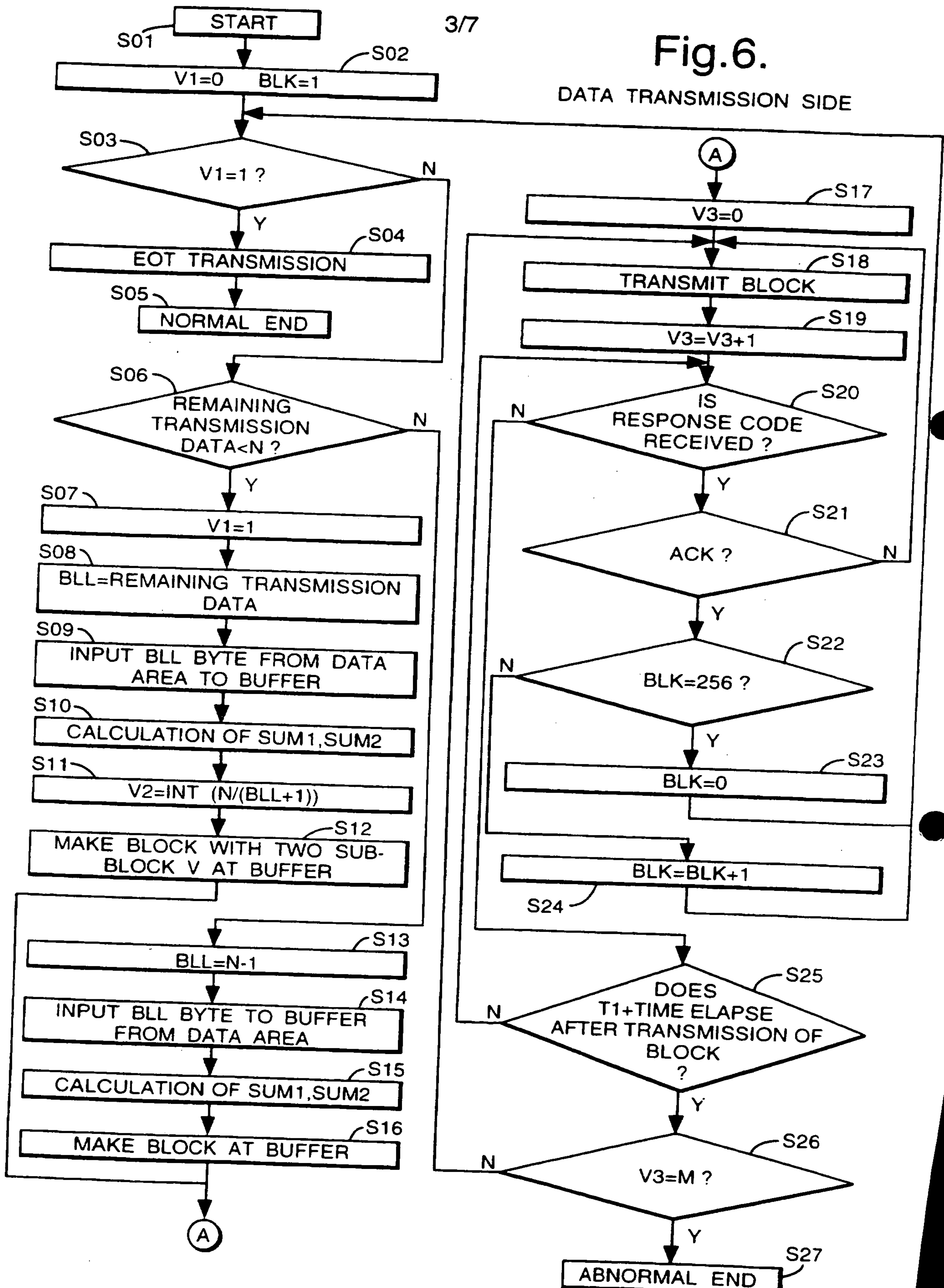


Fig.6.

DATA TRANSMISSION SIDE



XCID: <GB___2315984A__I_>

Fig.7.

DATA RECEIVING SIDE

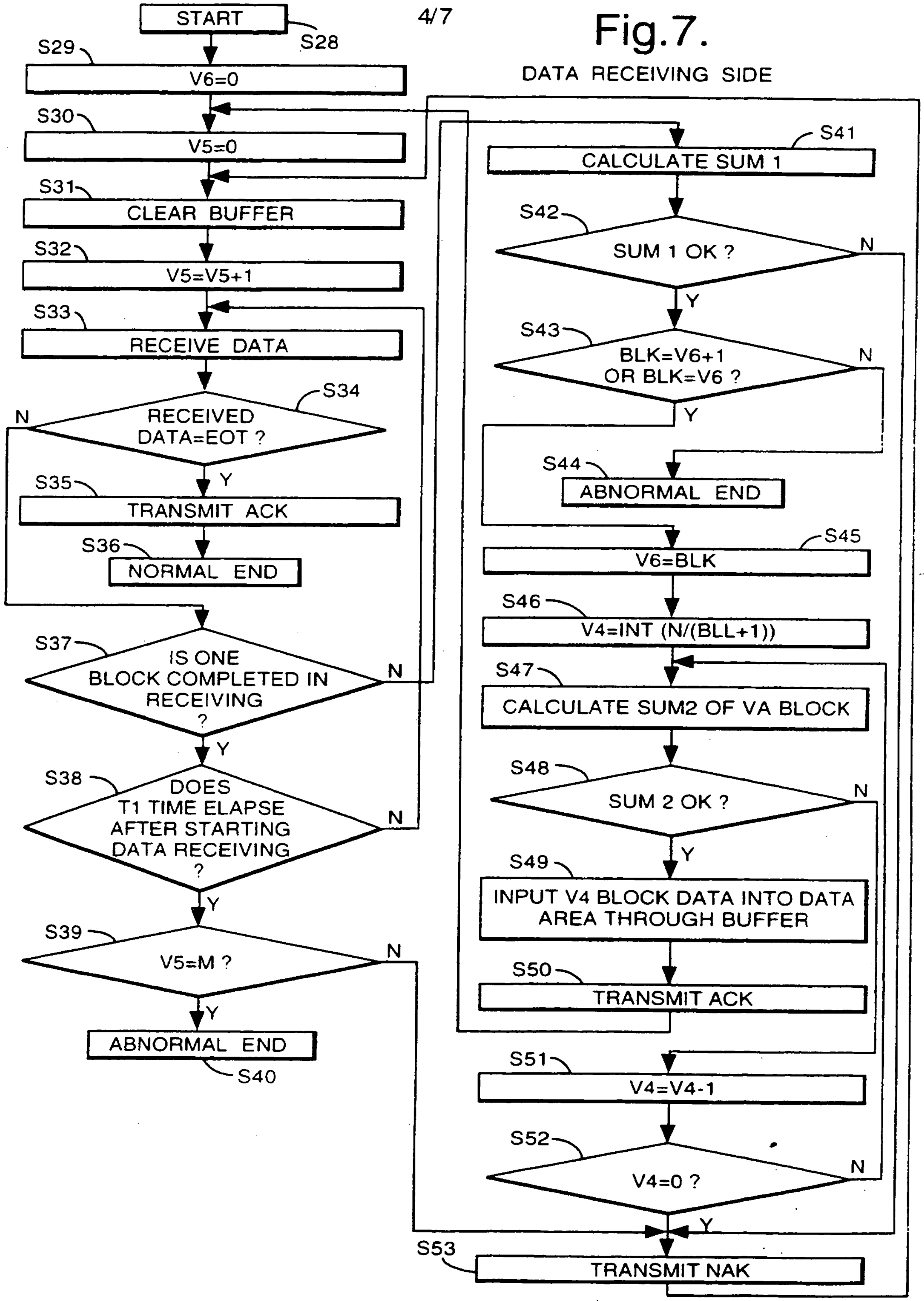


Fig.8.
PRIOR ART

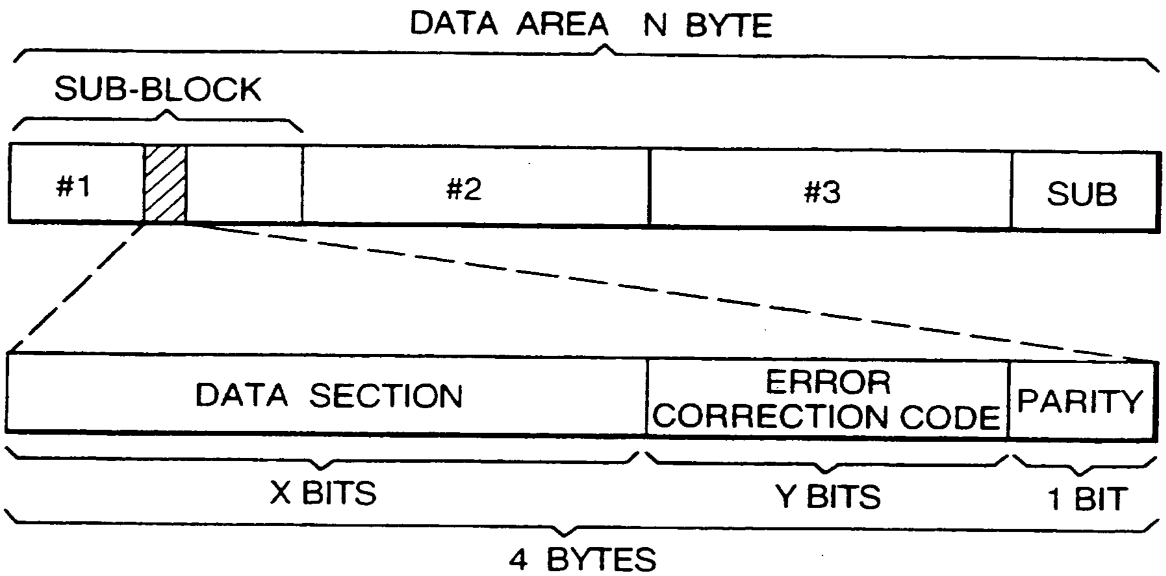


Fig.10.
PRIOR ART

SOH	BLK	$\overline{\text{BLK}}$	DATA	SUM

Fig.9A.

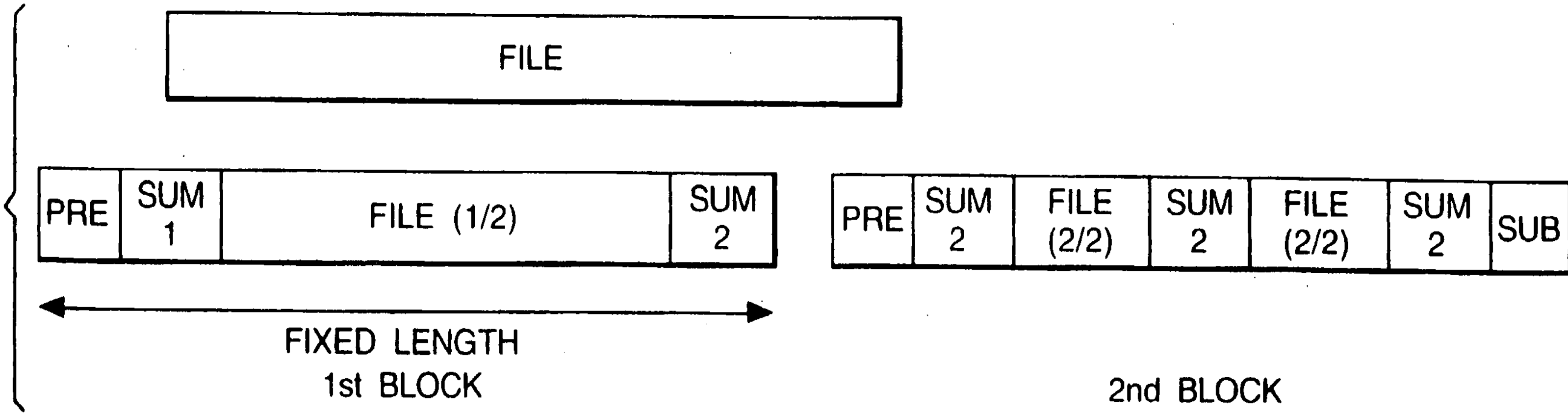

FILE
PRE
SUM 1
FILE (1/2)
SUM 2
PRE
SUM 2
FILE (2/2)
SUM 2
FILE (2/2)
SUM 2
SUB
FIXED LENGTH
1st BLOCK
2nd BLOCK

Fig.9B.

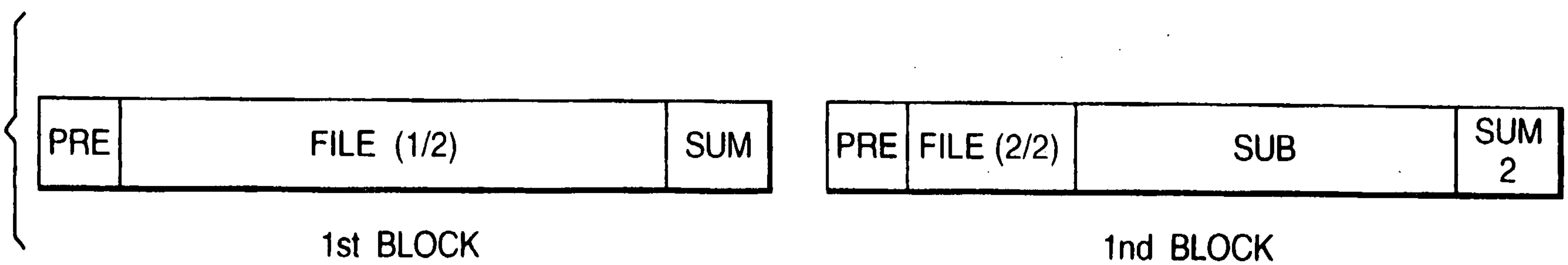

PRE
FILE (1/2)
SUM
PRE
FILE (2/2)
SUB
SUM 2
1st BLOCK
1nd BLOCK

Fig. 11.

PRIOR ART

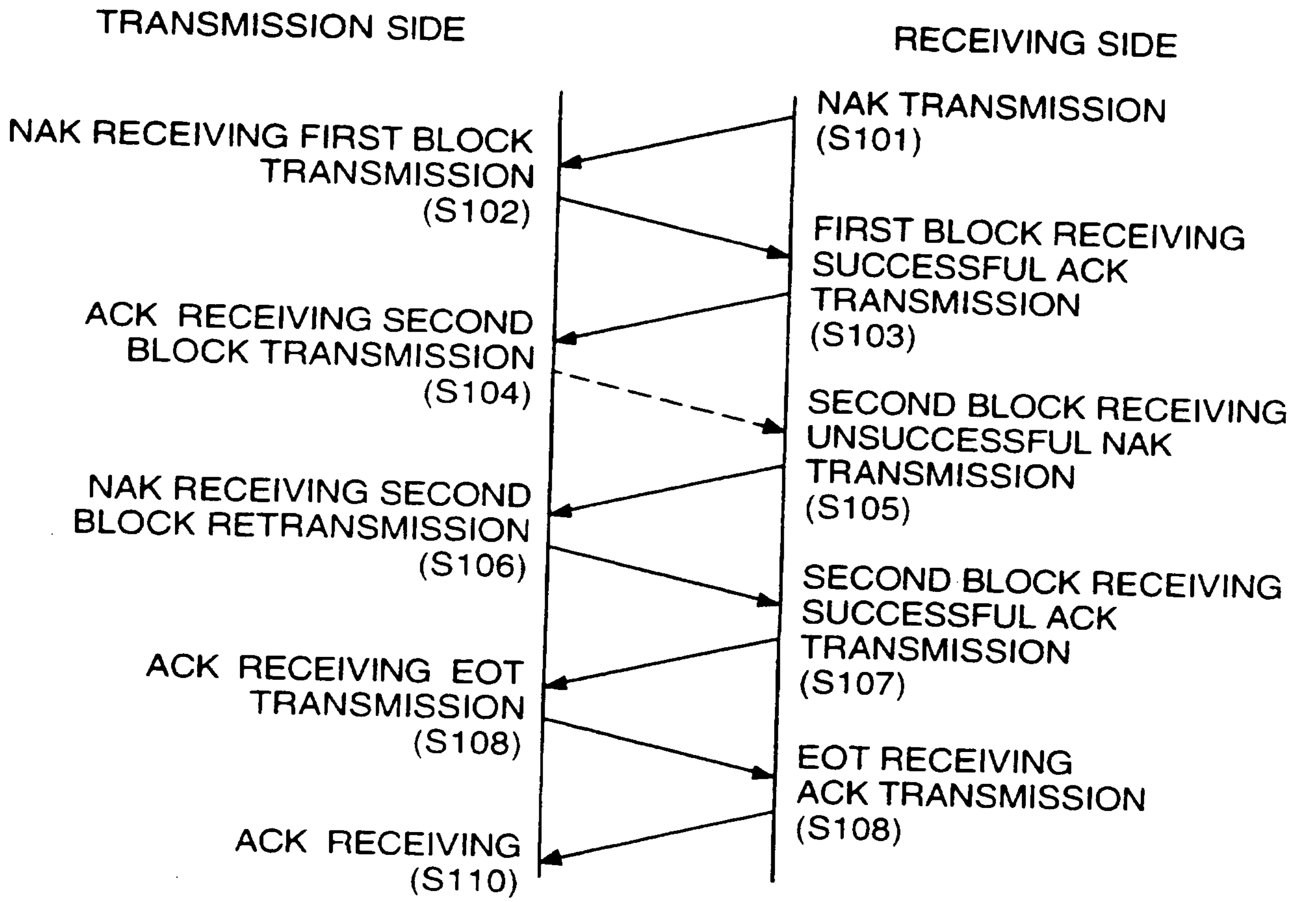
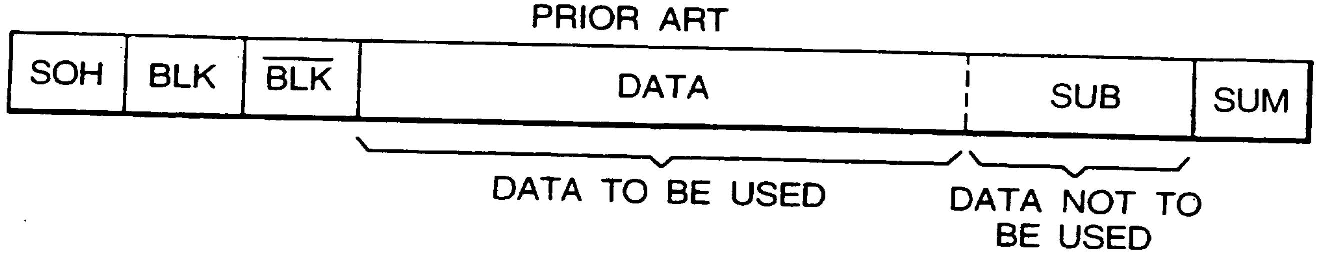


Fig. 12.

PRIOR ART



BSDOCID: <GB___2315964A__I_>

2315964

DATA COMMUNICATION SYSTEM AND METHOD THEREFOR

DESCRIPTION

BACKGROUND OF THE INVENTION

Field of the Invention

This invention relates to a data communication system and method therefor, and more particularly to a serial-type data communication system and method in which a fixed length block is used, and the block is divided into a plurality of sub-blocks in response to the length of each portion of data to be transmitted.

Description of the Related Art

A conventional data communication system is used for data transmission, or to control a data transmission device. The communication system operates over a wired (or wireless) communication system between, for example, personal computers or portable information terminals. Such data communication systems have been devised for various types of systems as described in "New Protocol Handbook" edited by Takashi Iida, published by Asahi Newspaper Co., Ltd. on October 5, 1994, pp. 200-206.

An Xmodem method is a basic method used in conventional data communication systems, wherein transmitted data is divided into fixed length blocks at the transmitting side, and all the blocks are separately transmitted to the receiving side.

Figure 10 illustrates the structure of a block used in the Xmodem method. A

BNSDOCID: <GB___2315964A_I_>

leading code (SOH) indicating a "start of heading", is placed at the leading end of the block. Next, a byte block number (BLK) indicates the logical order of the blocks (e.g., the order in which the blocks should be reassembled at the receiving side) and its complement (represented by a bar over BLK) is positioned after the BLK portion. After the block complement, a DATA portion having data situated therein, is provided. A checksum code (SUM) is located at the end portion of the block, and is for indicating whether the block contains an error.

Figure 11 schematically illustrates the data transmission side (and its operations) and the data receiving side (and its operations) in the Xmodem system.

First, the data receiving side (e.g., the right side in Figure 11) transmits an abnormal receiving code (NAK) (also referred to as a "negative acknowledgment") having a length of 1-byte (step S101) indicating that the receiving side is ready to receive a block (or another block) of data.

Then, the data transmitting side (e.g., the left side in Figure 11) transmits a first block of data (step S102). The format of the first block of data is the same as that shown in Figure 10 (or Figure 12 discussed below). The data receiving side receives this data, and checks the same for errors using the checksum code. Error-checking using a sum code is well-known to those ordinarily skilled in the art, and thus, for brevity, the details thereof will not be discussed herein.

When no error exists, the received data is stored, and a 1-byte normal received code (ACK) (also referred to as an "acknowledgment) is sent back to the transmission side (steps S103 and S107) to indicate an acknowledgment of receiving and storing the data. If an error is detected, the received data is canceled, and the NAK signal is sent back to the transmission side (step S105) from the receiving side.

The next block of data is transmitted from the transmission side to the receiving side when the next ACK signal is received (step S104). If the NAK signal is received, the same block is transmitted again (step S106) by the transmitting side until an ACK signal is received.

DOCID: <GB__2315964A__I_>

This procedure repeats, until the data transmission side transmits a 1-byte ending code (EOT) indicating an end of transmission, after the final block is transmitted (step S108). When the data receiving side receives the EOT signal, the receiving side issues an ACK signal to the transmission side, and the processing is completed (step S109).

In other communication systems, such as a Ymodem method/system or a Zmodem method/system, a more complex procedure is used. Such systems can accommodate large amounts of data and many files or the like.

However, there are several problems associated with conventional communication systems such as the Xmodem method/system.

First, when the length of the data to be transmitted is greater than the size of the block in a communication system using a fixed length block, such as the Xmodem system, multiple blocks and portions of blocks must be utilized. When portions of blocks are utilized, the remaining unused section (portion) of the block must be filled with a supplementary code (SUB) (also referred to as a "substitute code"), and must be transmitted as shown in Fig. 12. Similarly, when the length of the data to be transmitted is less than the fixed length of the block, the unused portions of the fixed length block must be filled in (e.g., completed) with a supplementary code. The supplementary code wastes space that could otherwise be used to transmit useful data, but it is necessary to allow the fixed length block to be transmitted and received integrally.

A second problem of the conventional communication systems is the potential use of long (e.g., extended length) blocks with fixed length block systems. A conventional fixed length block system using long fixed length blocks has a reduced transfer efficiency because such a system requires more frequent retransmission processing. The process of extending the length of the block may cause an error, resulting in a greater likelihood that a retransmission will be necessary. For example, the error results from transmission error due to lengthening of the blocks, and thereby resulting in an increased likelihood of

transmission noise. Generally, the longer the length of the block size, the more noise the block will encounter.

A third problem exists with fixed length block systems which reduce the block size (e.g., a system having small blocks). Such systems transmit data slowly when a large amount of data is transferred, because, when a system transmits with a smaller block size, the number of blocks required to transmit the same amount of data increases. Since each block must be formatted and acknowledged, the time needed to transmit a large number of small blocks is greater than the time needed to transmit a smaller number of large blocks, for a given quantity of data.

A fourth problem, for systems which use variable length blocks, is that the processing associated with such systems is very complicated. Communication systems using variable length blocks require a complex procedure for defining a block length between the transmission side and the receiving side, thereby decreasing communication speed and increasing the likelihood of errors.

Further, the Ymodem and Zmodem methods have problems. For example, the Ymodem method has many of the same problems as the Xmodem method described above, because it uses a fixed block length protocol similarly to the Xmodem method.

Further, while the Zmodem method utilizes a variable block length protocol and thus does not require supplementary code, it is a very complicated protocol and thus is undesirable.

Thus, the conventional methods/systems have many problems.

SUMMARY OF THE INVENTION

In view of the foregoing problems of the conventional systems and methods, an object of at least the preferred embodiments of the present invention is to provide a structure and method for a data communication system for effectively utilizing unused space within fixed blocks

DOCID: <GB___2315984A_I_>

while maintaining highly reliable data transmission.

In a first aspect, a serial data communication system for transmitting data according to the present invention, includes a transmitter for transmitting a fixed-length block of information and a receiver for receiving the fixed-length block from the transmitter. The fixed-length block includes a plurality of sub-blocks, each sub-block having a variable length depending upon the data.

With the unique and unobvious structure and method provided by the present invention, a data communication system is provided in which serial data communication is performed through a fixed-length data block, wherein when an amount of data to be transmitted is smaller than the size of the fixed-length block, the invention repeatedly arranges the transmission data within the block to completely utilize the fixed-length block. Thus, efficiency increases since there is no (or less) unused portion of the block, and the likelihood of errors decreases.

According to the invention, the data block may contain either a data error discriminating code (e.g., a checksum code or the like) or an error correction code. Normally, one code is sufficient (e.g., only one of the error cancelling systems is employed), since multi-error cancelling systems/methods are complicated and have poor efficiency.

Additionally, with the invention, data which does not contain errors (or contains errors which can be corrected) is preferentially selected. When such preferentially selected data is received by the receiving side, a response code is transmitted to the transmission side. Conversely, when the response code is not received by the transmission side, the data is retransmitted.

Therefore, with the invention, only a minimum amount of supplementary code is utilized, and either the data error discriminating code or an error correction code is added to the transmitted data. Then, at the data receiving side, data having no error (or an error which can be corrected) is preferentially selected from the received data block at the data receiving side. If the data contains an error, a request for retransmission of the block is performed.

BNSDOCID: <GB___2315964A__I_>

In another aspect of the invention, the probability of a block error in the second block is lower than conventional methods (e.g., an Xmodem method) since, when a data file is oversized, a second block includes a redundant data portion (e.g., a redundant "File (2/2)"). The second block of, for example, the conventional Xmodem method has no such redundant data portion, and an error is more likely as compared to the second embodiment of the present invention.

BRIEF DESCRIPTION OF THE DRAWINGS

The foregoing and other objects, aspects and advantages will be better understood from the following detailed description of a preferred embodiment of the invention with reference to the drawings, in which:

Figure 1 is a schematic block diagram of a data communication system according to the present invention;

Figure 2 is a schematic diagram of data block fields preferably used with the data communication system according to the present invention;

Figure 3 is a schematic diagram of a data area of a data block used with the data communication system according to the present invention;

Figure 4 is a schematic diagram of another example of a data area of a data block used with the data communication system according to the present invention;

Figure 5 is a schematic diagram of yet another example of a data area of a block used with the data communication system according to the present invention;

Figure 6 is a flowchart of an operation procedure occurring at a data transmission side of the data communication system according to the present invention;

Figure 7 is a flowchart of an operation procedure occurring at a data receiving side of the data communication system according to the present invention;

Figure 8 is a schematic diagram of a word used with the data communication

DOCID: <GB___2315964A__I_>

system according to the present invention;

Figure 9A illustrates a second embodiment of the present invention for transmission when data (e.g., a file) is larger than the block size, and, as a basis for comparison, Figure 9B illustrates such a transmission when using a conventional Xmodem system;

Figure 10 is a schematic diagram of a data block of the conventional Xmodem system;

Figure 11 is a schematic diagram of an operating procedure of the conventional Xmodem system; and

Figure 12 is a schematic diagram of a data block of the conventional Xmodem system.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS OF THE INVENTION

Referring now to the drawings, a preferred embodiment of the present invention will be described by way of example only.. Figure 1 illustrates a structure of a preferred embodiment of the present invention.

In Figure 1, a random access memory (RAM) 2 is divided into a data storing area 21 and a communication buffer area 22. The area 21 is for the data to be transmitted or received, so the size of the area is preferably not fixed. Preferably, the size of area 21 is dynamically adjusted. The area 22 is for the communication buffer so that size of this area is equal to the size of a block. Data to be transmitted and received is stored in the data storing area 21. The communication buffer area 22 is a temporary data storing memory capable of continuously transmitting or receiving data corresponding to one block.

A read-only memory (ROM) 3 stores a program for the data communication system, and includes a transmission software storing area 31 for storing transmission software and a receiving software storing area 32 for storing receiving

software.

A serial-parallel (S/P) data converter 4 converts parallel data used in a Central Processing Unit (CPU) 1 into serial data (used in serial communications), or to convert serial data into parallel data, respectively. Based on the programs contained in ROM 3, CPU 1 controls the data transmission operation among ROM 3, RAM 2 and S/P data converter 4, each connected together by a common system bus 5.

In a transmission operation, data is input to the data storing area 21 by an external input device (not illustrated). One block of the data within data storing area 21 is placed in the communication buffer area 22. It is noted that the block size is not fixed, but instead the user can select the block size suitably in advance based on design requirements and considerations. However, as known by one of ordinary skill in the art in the field, the transmitter and the receiver must choose the same size together in advance.

Transmission software within the transmission software storing area 31 processes the block of data within the communication buffer area 22, and outputs the data to the S/P converter 4. The converter 4 converts the data and outputs the data to the receiving side.

In a receiving operation, a block of data is received and converted by the S/P converter 4. Receiving software in receiving software storing area 32 controls the operation to store the block of data in communication buffer area 22. If the block of data is received without error (e.g., as determined by the checksum code or other error discriminating code such as CRC) or error correction code, an acknowledgment signal (ACK) is sent to the transmitting unit to allow a subsequent block to be sent. The blocks from the communication buffer area 22 are combined in the data storing area 21. Once the data transmission is complete, the accumulated data within the data storing area 21 is output to an output device (not illustrated).

Figure 2 is a block diagram illustrating an exemplary transmission data block used in the data communication system of the present invention.

DOCID: <GB___2315984A__I_>

First, a 1-byte leading code (SOH) is positioned at the leading end of the block to indicate the beginning ("start of heading") of the data block. In the next adjacent position, a one-byte block (BLK) indicates the logical order of the blocks (e.g., the numbered order in which all the blocks should be reassembled by the receiving side).

Adjacent to the BLK block is a one-byte data length code (BLL) which represents the length (e.g., the number of bytes) of the data being transmitted. In the above-described preferred embodiment, the length of data to be transmitted in 1 block is less than 256 bytes. Obviously, while, in the embodiment described above, the data area in the block has a maximum of 256 bytes, it is possible to set this value higher. The BLL variable could be multiple bytes in length. For example, if BLL has 2 bytes, the data area may be increased up to 65536 bytes.

A three-byte check sum code (SUM1) is positioned after the SOH, BLK and BLL bytes. N-bytes of data to be transmitted are positioned after the SUM1 code. With the arrangement discussed above, the total number of bytes in one block becomes N + 4 (e.g., SOH, BLK, BLL, SUM1), where N is the number of bytes of data to be transmitted. Specifically, N represents the fixed length of the data block.

Figure 3 illustrates the N-byte data area in greater detail. The data area includes data to be transmitted (DATA) in BLL bytes and a one-byte checksum code (SUM2).

The DATA and SUM2 fields form a sub-block, as shown in Figure 3. Therefore, the number of bytes in the sub-block is (BLL + 1) bytes. A number of bytes equal to (N - BLL - 1) bytes of the supplementary code (SUB) are inserted at the unused portion of the data area following the SUM2 field. Conversely, if the number of bytes in the sub-block is equal to the fixed length of the data area (i.e., (BLL + 1) = N), the supplementary code is not required, as shown in Figure 4.

Figure 5 illustrates an example of the block configuration where the size of the data length code BLL (e.g., for each sub-block) is at least two bytes smaller than

one-half the size of the data area (i.e., BLL $\leq$ (N - 2)/2). For example, assuming N (total bytes) is equal to 6, if DATA (BLL) is equal to 3, the equation is not satisfied since (6-2)/2 is equal to 2. However, if N is equal to 6 and DATA is equal to 2 (e.g., for each sub-block), then the equation is satisfied. For DATA (BLL) equal to 3, N would be equal to 8. Such a configuration allows a plurality of sub-blocks to be placed in a single data area such that the number of bytes of supplementary code (SUB) is minimized.

The check sum codes are used to verify the data transmitted in a conventional manner as is known by those ordinarily skilled in the art. Specifically, the check sum codes are provided to detect an error in the data. For example, SUM1 is selected such that a summation of the total value of SOH + BLK + BLL + SUM 1 is equal to 0. The reason is that such a summation identifies the presence (e.g., when not equal to 0) or absence (e.g., when equal to 0) of errors in transmission.

Similarly, when the value of SUM2 is one byte less than a total value of each of the data in the data area, SUM2 becomes 0. This is a conventional check sequence. SUM2 is a check sum code for the data in the data area (e.g., total BLL bytes).

Hereinbelow, the operation of the preferred embodiment of the present invention is described with reference to the flowcharts of Figures 6-7, which illustrate the operations at the data transmission side and at the data receiving side, respectively.

Operation at the Data Transmission Side

With regard to the transmission side operations, referring to Figure 6, the procedure begins in step S01, and thereafter an applied variable V1 is initialized to 0, and BLK is initialized to 1 (step S02). As discussed in further detail below, when the last data transmitted has a length less than N bytes (e.g., when the block is a final block), V1 is set to 1. Otherwise, V1 is 0. As discussed above, BLK is the

BSDOCID: <GB___2315964A__I_>

block order number and is added to each block.

In step S03, it is determined whether V1 is 1. If V1 is 1, an end of transmission signal (EOT) is transmitted (step S04) and the processing finishes normally (step S05).

If the value of V1 is 0 (e.g., a "NO" in step S03), then processing continues to step S06. In Step S06, it is determined whether the remaining transmission data (e.g., data block(s) remaining of the total data to be transferred) is less than N (e.g., the length of the data is less than the size of the data area (N bytes)) (step S06). More specifically, subsequent blocks are processed until the last block. The final block is then examined as to whether it is less than N.

When the remaining transmission data is less than N bytes, that block becomes a final block, so that the variable V1 is set to 1 (step S07), and the variable BLL is set to the remaining byte number of the transmitted data (step S08). Then, the BLL byte is output from the data area (e.g., data storing area 21 of Figure 1) to the buffer (e.g., communication buffer area 22 of Figure 1), and is used for calculating SUM1 and SUM2 (at step S10). The BLL byte affects the check sum values, as described above

The invention further reduces (e.g., minimizes) the number of supplementary codes (SUB) required by determining how many sub-blocks can be placed in a single data area.

Specifically, a variable V2 is set equal to the integer quotient of N/(BLL + 1) (e.g., {INT (N/(BLL + 1)}) (step S11). Then, the block is formed with multiple sub-blocks in the buffer 22 (step S12).

Conversely, in step S06, when the remaining value is more than N bytes, BLL is initially set to the number of bytes in the data area (e.g., N - 1; step S13). The BLL byte is input from the data area 21 to the buffer 22 (step S14), and is used to calculate SUM1 and SUM2 (step S15). Then, the block is formed from the information within the buffer 22 (step S16).

Upon completion of the block configuration, the block is transmitted. A

variable V3 which is used to count the number of retransmission times is set to 0 (step S17). Such a feature can be implemented as an incremental counter either in hardware and/or software, and is believed to be well-known to one of ordinary skill in the art. Thus, for brevity, the structure and operation thereof will not be discussed in detail herein.

Then, one block is transmitted (step S18), and V3 is incremented by one (step S19). Then, the transmission side waits a predetermined time T1 to receive a response code from the receiving side (step S20).

If the receiving side returns an ACK code to the transmission side (step S21) and if the value of BLK is 256 (step S22), the BLK is set to 0 (step S23).

Conversely, if the receiving side returns an ACK code to the transmission side (step S21) but the value of BLK is not 256 (as determined in step S22), BLK is incremented by 1 (step S24) and the operation returns to step S03 for further processing. When the response code is not an ACK code, the operation returns to step S18 for a block retransmission operation.

If a response code from the receiving side is not detected in step S20, but the predetermined time period T1 following the transmission has not elapsed (step 25), the operation returns again to the response code receiving determination step (step S20). If time T1 has elapsed but there have not been M (e.g., 3) attempted transmissions (step S26), then the processing returns to the transmitting operation (step S18) to retransmit the block. If the time T1 elapses and there have been M (e.g., 3) attempts at transmission of the same block, then the processing finishes in an abnormal state (step S27) and is deemed to be abnormally terminated.

Thus, the time period T1 is a time for waiting for the response code from the receiving side. If the response code is not detected when T1 elapses, a block retransmission is performed. The variable M is the maximum allowable number of times for block transmission (e.g., set to 3 in the above example; specifically, the block is transmitted once and then retransmitted up to two more times for a total of 3 block transmissions). When the response code is not detected and the

ISDOCID: <GB_____2315964A_I_>

transmission is performed M times, the processing terminates abnormally.

The variables T1 and M are predetermined values and are set in advance at the transmission side and the receiving side, respectively, depending upon the designer's requirements and constraints. Thus, T1 and M may be any suitable values as determined by the designer taking into consideration the designer's requirements.

Operations at the Data Receiving Side

Referring to the flowchart in Figure 7, which illustrates the operation at the data receiving side, the process begins at step S28. Initially, variables V6 and V5 are initialized to 0 (steps S29 and S30). Variable V5 is used to store the number of times the response code is transmitted, and V6 is used to store the block number of the previously received block.

The receiving buffer 22 is cleared (step S31), and the variable V5 is incremented by 1 (step S32). Then, the operation waits for the detection of data transmitted from the transmission side (step S33) (e.g., the data receiving side waits to receive data).

When a data transmission is detected, the first byte of the transmission is checked to determine if it is an EOT code (step S34). If the byte is an EOT code, the receiving side transmits an ACK code (step S35) to the transmission side, and the processing completes normally (step S36).

Conversely, in step S34 if it is determined that the first byte is not the EOT code and data is not detected, a receiving operation is performed, and it is determined whether a full block (N + 4 bytes) has been received (step S37). Such a determination may be made, for example, by the receiving side having the same EOT code as the transmitting side and performing a comparison upon receipt of the transmission. If the first byte is not an EOT, the transmitter will transmit a full block, so that the receiver will receive a ful block.

In the event that receiving the one block is not completed, the time period T1

BNSDOCID: <GB___2315964A__I_>

is measured from when the response code is transmitted (step S38) (e.g., it is determined whether T1 elapses after starting the data receiving operation). If T1 has not elapsed, the operation subsequently returns to the receiving operation (step S33) to receive data. When T1 elapses, the variable V5 is compared with a predetermined value M (step S39). M is the maximum allowable number of times the response code can be transmitted.

If V5 is less than M, a negative acknowledgment (NAK) is sent from the data receiving side to the transmission side (step S53), and the operation returns again to the buffer clearing operation (step S31) to again perform a receiving operation. If T1 elapses and V5 is equal to M, the processing is stopped abnormally (step S40) (e.g., an "abnormal termination").

The time period T1 is for waiting for the data block from the transmission side. If the data block is not received within the time period T1, the inventive method again performs a response code transmitting operation (e.g., a re-transmitting operation). When the data block is not received within time period T1 and the acknowledgment code is transmitted M times, the processing is terminated and judged to be an abnormal termination. The values T1 and M are predetermined values, and are set in advance at both the transmission side and the receiving side. It is noted that T1 and M are the same at both the transmission side and the receiving side.

When a block (e.g., one block) has been received, SUM1 is calculated (step S41) and it is determined whether SOH, BLK and BLL are correct (step S42) by using the summing equation described above (e.g., determine whether SOH + BLK + BLL + SUM1 = 0). If SOH, BLK and BLL are not correct, all data in this block is not reliable. Thus, preferably it is determined first whether these codes are correct.

If an error is detected in step S42, a negative acknowledgment (NAK) is transmitted from the receiving side to the transmission side (step S53), and the operation returns to the buffer clearing operation (step S31) and to perform again

INSDOCID: <GB___2315964A_I_>

the receiving operation.

If SUM1 is normal (e.g., "OK" in step S42), then variable BLK is checked to determine whether BLK = V6 or BLK = V6 + 1 (step S43). If BLK = V6, then it is determined that the data block is a retransmitted data block. If BLK = V6 + 1, then it is determined that the block is a next block subsequent (adjacent) to the previously received block.

Since either of the above two determinations represents normal procedures, the variable V6 is set to BLK (step S45), and the operation advances to the next step. However, if BLK is other than V6 or V6 +1, it is determined that a contradiction (e.g., error) in the communication procedure has occurred, and the operation is terminated abnormally (step S44).

In step S46, the number of sub-blocks included in the received block is calculated with reference to the integer quotient of N/(BLL + 1), and the integer quotient is substituted for the variable V4. Then, the SUM2 in the V4 block is calculated (e.g., usually it includes 1 byte but it could be larger). For example, as shown in Figure 4, N (bytes) - BLL (byte) = SUM2 (step S47). This operation is significant in that the calculation is for obtaining how many sub-blocks are in the block. After this calculation, the receiver starts to search for a correct sub-block.

Then, the data in the sub-block is checked for correctness by checking each sub-block by using SUM2 (step S48).

If an error is not detected (e.g., SUM2 is "OK" as determined in step S48), data in the sub-block is stored in the data storing area 21 (step S49). If the block is determined to be the same as the previous block, the previous data is replaced with the current data transmission. If the data is determined to be the next block (e.g., adjacent to the previously sent block), it is positioned following (subsequent to) the data received previously. After this operation, an acknowledgment (ACK) is transmitted to the transmission side (step S50), and the operation returns to the processing for receiving the next block (step S30).

When an error is detected (e.g., SUM2 is determined not to be "OK" in step

S48), V4 is reduced by 1 (step S51), and a similar operation is performed for the next sub-block. This operation is repeated until V4 = 0 and a determination is made as to whether the data of all the sub-blocks are correct (step S52).

When the data in one or more of the sub-blocks is incorrect, a negative acknowledgment (NAK) is transmitted by the receiving side to the transmission side (step S53), and the operation returns to the buffer clearing operation (step S31) and performs another receiving operation.

If the transmission side and the receiving side are operated together in accordance with the aforesaid procedure and both the transmitting side and the receiving side terminate normally, it is judged that the data has been transferred correctly to the receiving side.

While, in the preferred embodiment described above, the checksum has been utilized as the data error discriminating code for each sub-block, it is also possible to use a data error correcting code. For example, a method may be provided in which the data error correcting code is divided into a data section (X bits), an error correcting code section (BCH code) (Y bits), and a parity bit section (1 bit) in a unit of 1 word (4 bytes) could be included in the sub-block, as shown in Figure 8. If the data section X = 21 bits and the error correction code Y = 10 bits, a data error of 2 bits may be corrected in the data section. Since use of the error correcting codes (ECC) is well-known, for brevity it will not be described in detail herein.

Specifically, in the data error correction procedure, first the transmission data is converted into a word (e.g., as shown in Figure 8) at the transmission side so as to represent the sub-block with this word.

If an error exists, then the receiving side performs an error correction on the sub-block based on the word, and recovers the otherwise erroneous data. If the error correction cannot be performed (e.g., a word is contained in the sub-block in which error correction cannot be performed), then the sub-block is not usable. As discussed above, if one of the sub-blocks includes an error, a request to retransmit the data will be sent to the transmission side. Thus, as long as the data in all the

NSDOCID: <GB__2315964A__I_>

sub-blocks repeatedly arranged in one block cannot be used, a request for re-transferring is performed with respect to the transmitting side.

As can be understood from the foregoing description, according to the data communication system of the present invention, a highly reliable and more efficient communication procedure can be performed, even when the data to be transmitted has a length smaller than the fixed block length. Further, the invention uses a simple communication protocol associated with the fixed length block communication. The inventive method and structure improve communication response and reduce power consumption.

Specifically, the present invention is much less complicated than the variable length block method described in the background section. Generally, the protocol using variable length block needs BOF (Beginning of Field), EOF (End of Field) and CE (Control Escape) and so forth. Processing these codes is complicated. The invention provides a much simpler and easier protocol.

Since multiple sub-blocks are present in one block, a retransmitting operation is unnecessary so long as the data in all the sub-blocks are free of error, thereby resulting in a reduction of the number of fixed length block transmissions.

Second Embodiment

It is noted that, while the foregoing describes a situation where the block size is less than one-half the length of the fixed-length data field, it also could be applied to an oversized block size which is larger than the fixed-length data field.

Specifically, a second embodiment of the invention deals with such a situation in which a file which is larger than the block size, as shown in Figure 9A. In contrast, the conventional Xmodem method treats such a situation as shown in Figure 9B. Such Figures clearly show the advantages of the second embodiment of the present invention.

In Figure 9A, in the first block the block has a fixed length, and utilizes PRE and SUM1 at the beginning of the block and SUM2 at the end of the block. The

data (e.g., "File (1/2)") is therebetween. The PRE section includes SOH + BLK + BLL.

In the second block of Figure 9A, the transfer protocol is shown which includes PRE and SUM1 before a first file (e.g., "File (2/2)") portion and SUM2 immediately thereafter, followed by a second (redundant) "File (2/2)" portion and another SUM2 code. The transmission is completed with a SUB code. Thus, the second block of Figure 9A includes a double (redundant) "File (2/2)", and hence the probability of a block error is lower than the conventional Xmodem method illustrated in Figure 9B.

Specifically, in Figure 9B showing the conventional Xmodem system, a PRE code and a SUM code are at first and second ends, respectively, of a "File (1/2)" portion of a first block. In a second block, a PRE code is followed by a "File (2/2)" portion, followed by a large SUB code portion and a SUM code.

Thus, with the second embodiment of the present invention, the probability of a block error in the second block is lower since the second block of 9A includes a redundant "File (2/2)". In contrast, the second block of the conventional Xmodem method, as illustrated in Figure 9B, has no such redundant "File (2/2)", and thus an error is more likely.

While the invention has been described with reference to preferred embodiments described above, it is not limited thereto and includes all variations falling within the claims.

Each feature disclosed in this specification (which term includes the claims) and/or shown in the drawings may be incorporated in the invention independently of other disclosed and/or illustrated features.

INSDOCID: <GB___2316964A__I_>

The text of the abstract filed herewith is repeated here in full as part of the specification.

A data communication system and method for transmitting serial data, includes a transmitter for transmitting a fixed-length block of information, and a receiver for receiving the fixed-length block from the transmitter. The fixed-length block includes a plurality of sub-blocks, and the plurality of sub-blocks each have a variable length depending upon the data. When an amount of transmission data has a size less than that of the fixed-length data block at a transmitter side, the information is repeatedly arranged within a same block in a range not exceeding the block size so as to form the block. Thus, in the data communication system having a fixed-length data block, the unused portion of the fixed-length block is effectively utilized in a highly reliable data communication. If the data transmitted has a size less than that of the data area of the fixed-length data block, multiple sub-blocks and an error check code are stored within the data area. With such an arrangement, error-free data is preferentially received so as to improve reliability and data transmission efficiency.

3NSDOCID: <GB___2315964A__I_>

CLAIMS

1. A data communication system for transmitting serial data, comprising:

a transmitter for transmitting a fixed-length block of information; and

a receiver for receiving said fixed-length block from said transmitter,

wherein said fixed-length block includes a plurality of sub-blocks, and said plurality of sub-blocks each have a variable length depending upon said data,

wherein when an amount of transmission data has a size less than that of said fixed-length data block at a transmitter side, said information is repeatedly arranged within a same block in a range not exceeding said block size so as to form said block.

2. The data communication system as set forth in claim 1, wherein said fixed-length block includes one of a data error discriminating code and an error correcting code.

3. The data communication system as set forth in claim 2, said receiver further comprising means for correcting an error based on said error correcting code, said receiver preferentially accepting data free of error.

4. The data communication system as set forth in claim 3, wherein when said receiver accepts error-free data, said receiver transmits to said transmitter a response code representing that the data is received.

5. A data communication system as set forth in claim 4, wherein when said

BSDOCID: <GB___2316984A_I_>

response code is not received by said transmitter, said data is transmitted again to said receiver.

6. The data communication system according to claim 1, wherein said transmitter includes:

means for determining whether a response code has been received from said receiver after transmission of data; and

means for determining whether a predetermined time period has elapsed after said transmission of data, wherein, after said predetermined time period has elapsed and said means for determining determines that no response code has been received, said data is transmitted again to said receiver.

7. A data communication system according to claim 1, wherein said receiver requests said transmitter to retransmit a received block only when each sub-block contained in the received block contains an error.

8. A method for serial data communication, comprising steps of:

determining a maximum number of sub-blocks that a fixed-length block can accommodate;

storing at least two said sub-blocks in said fixed-length block; and

transmitting said fixed-length block.

9. The method as in claim 8, further comprising a step of:

determining a size of said fixed-length block.

10. The method as in claim 8, further comprising a step of:

storing one of a data error discriminating code and an error correcting code in said fixed-length block before said transmitting step.

11. The method as in claim 8, further comprising steps of:

receiving said fixed-length block;

checking said fixed-length block for an error; and

if said fixed-length block contains an error, correcting said error.

12. The method as in claim 8, further comprising steps of:

receiving said fixed-length block;

checking said fixed-length block for an error; and

if said error cannot be corrected, requesting a retransmission of said fixed-length data block.

13. The method as in claim 8, further comprising steps of:

receiving said fixed-length block;

checking said fixed-length for an error;

if said fixed-length does not contain error, transmitting an acknowledgment representing an error-free receipt of said fixed-length block; and

retransmitting said fixed-length block if said acknowledgment is not transmitted.

14. The method according to claim 8, wherein when an amount of transmission data has a size less than that of said fixed-length data block at a transmitter side, said information is repeatedly arranged within a same block in a range not exceeding said block size so as to form said block.

15. The method according to claim 8, further comprising steps of:

determining whether a response code has been received from a receiver side after transmission of data; and

determining whether a predetermined time period has elapsed after said transmission of data, wherein, after said predetermined time period has elapsed and

NSDOCID: <GB___2315964A_I_>

it is determined that no response code has been received, said data is re-transmitted.

16. The method according to claim 8, further comprising steps of:

receiving said fixed-length block as a received block at a receiving side;

determining whether each sub-block in said received block includes an error; and

retransmitting a received block only when each sub-block contained in the received block contains an error.

17. A data communication system for communicating serial data, comprising:

a transmitter including means for arranging said data in a fixed-length data block; and

a receiver for receiving said data transmitted,

wherein when the data transmitted has a size less than that of the data area of the fixed-length data block, a plurality of sub-blocks and an error checking code are stored within the data area.

18. The data communication system according to claim 17, wherein said receiver requests said transmitter to retransmit a received block only when each sub-block contained in the received block contains an error.

19. A data communication method or a data communication system substantially as herein described with reference to any of Figures 1 to 9A of the accompanying drawings.

BNSDOCID: <GB 2315984A I >

The Patent Office

Application No: GB 9716206.9

Claims searched: 1-7

Examiner: Stephen Brown

Date of search: 30 October 1997

Patents Act 1977
Search Report under Section 17

Databases searched:

UK Patent Office collections, including GB, EP, WO & US patent specifications, in:

UK Cl (Ed.O): H4P (PENL, PENX)

Int Cl (Ed.6): H04L: 1/00, 1/08, 1/18, 29/06, H03M: 7/40.

Other: Online: WPI, JAPIO.

Documents considered to be relevant:

Category	Identity of document and relevant passage	Relevant to claims
A	EP 0 503 768 A1 (IBM)	-
A	US 5 410 308 (Thomson-Brandt)	-
A	US 5 079 548 (Fujitsu)	-

X	Document indicating lack of novelty or inventive step	A	Document indicating technological background and/or state of the art.
Y	Document indicating lack of inventive step if combined with one or more other documents of same category.	P	Document published on or after the declared priority date but before the filing date of this invention.
		E	Patent document published on or after, but with priority date earlier than, the filing date of this application.
&	Member of the same patent family		